Measuring Risks of Heart Disease Using Logistic Regression

Tziporah Horowitz

Johns Hopkins University

Whiting School of Engineering

Applied and Computational Mathematics 725

December 13, 2022

Abstract

Heart disease describes a range adverse conditions that affect the heart such as, coronary artery disease and heart attack. It is the leading cause of death in the United States. While some people may be genetically predisposed toward developing heart disease, others can have an increased risk due to poor diet and exercise and other biological conditions. Knowing the association between risk factors and the chances of developing heart disease can allow one to take preventative action by changing lifestyle habits. This paper discusses statistical methods that can be used to measure the associated risk between various risk factors and the likelihood of developing heart disease. An analysis was performed using real-world data to compare simple methods like risk difference and relative risk to those that can be measured by a logistic regression model. After determining the associated risks, the logistic regression model's predictive capabilities were tested, resulting in high accuracy.

1 Introduction

Statistical binary classification is a process used in medical testing to determine whether or not a patient has a certain disease. It often involves supervised learning to determine decision classes based on predefined rules. While methods such as decision trees, random forests, and neural networks are commonly used in other applications of binary classification, logistic regression offers medical testing a number of advantages due to its probabilistic nature and its relationship with the odds ratio (Schober and Vetter, 2021). If interpreted correctly, it can provide a measure of associated risk between the independent variables and the binary outcome.

The upcoming sections will provide an overview of the statistical theory behind logistic regression, followed by an analysis of the association between various risk factors and the chances of a patient developing heart disease. To measure the associated risk, this paper provides a synopsis of four common methods: *risk difference*, *relative risk*, *odds ratios*, and *marginal effects*, two of which are illustrated using a logistic regression model.

2 Overview of Logistic Regression

Logistic regression was proposed as an alternative to ordinary least squares (OLS) regression in the 1960's in the context of predicting binary outcomes (Peng et al., 2002). While OLS relies on linearity, normality, and continuity, logistic regression utilizes the *logit* or log-odds function (eq. 1) to predict the probability of an outcome falling into a specific category.

$$\text{logit}\,(Y) = \ln\left(\frac{p}{1-p}\right) \tag{1}$$

Using the logit function allows the modeler to create a sigmoidal relationship between two classes, which appears linear in the middle and curved on the ends.

Let $\hat{p}$ be the probability of an outcome occurring given a specific value of a feature:

$$\hat{p} = \mathbb{P}(Y = 1|X = x) = \frac{1}{1 + e^{-(\alpha+\beta x)}} \tag{2}$$

By rewriting the sigmoid function in equation 2 and taking its natural logarithm, we can derive a linear relationship between the log-odds of $\hat{p}$ and the feature variable, x:

$$\begin{aligned}
\hat{p} &= \frac{1}{1 + e^{-(\alpha+\beta x)}} \\
\frac{1}{\hat{p}} &= 1 + e^{-(\alpha+\beta x)} \\
\frac{1}{\hat{p}} - 1 &= e^{-(\alpha+\beta x)} \\
\frac{1-\hat{p}}{\hat{p}} &= e^{-(\alpha+\beta x)} \\
\ln\left(\frac{1-\hat{p}}{\hat{p}}\right) &= -(\alpha + \beta x) \\
\ln\left(\frac{\hat{p}}{1-\hat{p}}\right) &= \alpha + \beta x
\end{aligned}$$

Thus, logistic regression can be expressed a generalized linear model (GLM), such that

$$\text{logit}(Y) = \vec{\beta} X$$

where $\vec{\beta}$ is the vector of regression coefficients and X is the matrix of feature variables.

While there are a number of different techniques, the regression coefficients are typically estimated using the *maximum likelihood* method. The maximum likelihood method aims to maximize the likelihood of reproducing the data given the parameter estimates (Peng et al., 2002). Let

$Y_i|X_i \overset{ind}{\sim} \text{Bernoulli}\,(f(\vec{x}_i))$. Then, the joint likelihood function is:

$$\mathscr{L}\left(\vec{\beta}\right) = \mathbb{P}\left(Y_1 = y_1, \ldots, Y_n = y_n | \vec{x_1}, \ldots, \vec{x_n}\right) = \prod_{i=1}^{n} f(\vec{x}_i)^{y_i} \left(1 - f(\vec{x}_i)\right)^{1-y_i}$$

Thus, the maximum likelihood estimate of the logistic regression function is:

$$\vec{\beta} = \underset{\vec{w}}{\text{argmax}} \left\{ \prod_{i=1}^{n} \left(\frac{1}{1+e^{-\vec{w}X}} \right)^{y_i} \left(\frac{1}{1+e^{\vec{w}X}} \right)^{1-y_i} \right\} \tag{3}$$

However, it is important to note that the gradient vector of equation 3 cannot be solved for zero, and therefore has no closed form. A common approach for solving equation 3 numerically is the *reweighted least squares* algorithm (Wasserman, 2004).

The coefficients of a logistic regression model can be interpreted as, β_i is the change of the log-odds of the target occurring per one unit increase of X_i when all other variables X_j $(j \neq i)$ are fixed. The estimates, $\hat{\beta}_i$ can be tested for statistical significance using a Wald test where

$$H_0 : \hat{\beta}_i = 0$$

$$H_1 : \hat{\beta}_i \neq 0$$

If $|W| = \left| \frac{\hat{\beta}_i - 0}{\hat{\text{se}}} \right|$ is greater than $z_{\frac{\alpha}{2}}$, then the null-hypothesis is rejected and $\hat{\beta}_i$ is considered statistically significant.

3 Methods

There are several ways to measure the association between a risk factor and the binary outcome of contracting a disease. The following sections discuss the four approaches used in this analysis to determine the associated risk of various factors in the context of patients contracting heart disease.

	Diseased	Healthy
Exposed	D_E	H_E
Unexposed	D_U	H_U

Table 1: Contingency Matrix

Two of the methods are computed algebraically, while the other two can be derived from a logistic regression model. Sections 3.1 through 3.3 refer to table 1 for simplification.

3.1 Risk Difference

Often considered the simplest approach for measuring associated risk, *risk difference* or *absolute risk difference* (ARD) is the difference in the outcome rates between patients with the risk factor and patients without the risk factor (Telke and Eberly, 2011). Using the matrix in table 1, risk difference can be defined mathematically as:

$$\text{ARD} = \frac{D_E}{D_E + H_E} - \frac{D_U}{D_U + H_U}$$

While the risk difference is easy to compute, its interpretation is often misleading and can only explain the associated risk between a single factor and the target.

3.2 Relative Risk

Similar to risk difference, *relative risk* compares the outcome rates between patients with the risk factor and patients without the risk factor. However, relative risk is computed as a ratio (RR) rather than a difference (Telke and Eberly, 2011). The risk ratio is defined as:

$$\text{RR} = \frac{D_E/(D_E + H_E)}{D_U/(D_U + H_U)}$$

Relative risk is a useful statistic because it quantifies the probability of a patient with exposure contracting the disease relative to a patient without exposure. Risk ratios that are close to 1 indicate that the risk of contracting the disease for an exposed patient is the same as the risk for an unexposed patient. In contrast, risk ratios that are far from 1 indicate that there is an association between the variables. This allows one to create a confidence interval using the hypothesis test,

$$H_0 : RR = 1$$

$$H_1 : RR \neq 1$$

The risk ratio is considered a valid measure of relative risk for studies in which the sampling is dependent on the exposure of interest such as, randomized controlled trials or cohort and cross-sectional studies (Gallis and Turner, 2019). Like risk difference, relative risk can only explain the associated risk between a single factor and the target.

3.3 Odds Ratio

Often confused with risk ratio, *odds ratio* compares the statistical odds of the outcome in the exposed group to that of the outcome of the unexposed group. It is defined mathematically as:

$$\text{OR} = \frac{D_E/H_E}{D_U/H_U}$$

Like the risk ratio, odds ratios that are close to 1 indicate no association between exposure and contracting the disease, and odds ratios that are far from 1 indicate that there is an association between the variables. One can also create a confidence interval for the odds ratio using a similar

hypothesis test to that of the risk ratio, such that

$$H_0 : OR = 1$$

$$H_1 : OR \neq 1$$

While the odds ratio is typically considered the "only valid measure of relative association in traditional case-control studies" (Gallis and Turner, 2019), it is frequently misinterpreted as the risk ratio. However, in cases where the risk factor is relatively small ($< 10\%$), the odds ratio approximates the risk ratio:

$$\frac{D_E/(D_E+H_E)}{D_U/(D_U+H_U)} \approx \frac{D_E/H_E}{D_U/H_U} \qquad \text{if } D_E \text{ and } D_U \text{ are small.}$$

The odds ratio can be applied in multi-parameter settings when computed in a logistic regression analysis, due to its inherent calculation of the logit (or log-odds) function. To obtain the odds ratio of a logistic regression model, one simply has to exponentiate the coefficients.

3.4 Marginal Effects

A *marginal effect* (or incremental effect) is the change in the probability that an outcome occurs as the risk factor changes by one unit. It is often used in logistic regression analysis and other GLM's to explain the incremental risk associated with each factor (Norton et al., 2019). Unlike the odds ratios, the marginal effects are not a function of the regression coefficients. Marginal effects are determined by taking the partial derivative of the regression equation with respect to each variable. They are simpler to interpret than odds ratios and are easier to compare across different studies. There are many ways to represent the marginal effect for a sample, the most common of which is the *average marginal effect* across all patients in the dataset.

3.5 The Analysis

For this analysis, the *Heart Disease Dataset* was collected from kaggle.com (Lapp, 2019). The dataset includes data that was compiled from four databases in 1988 and consists of 1025 rows and 14 columns: 13 predictors and 1 target. The predictors include 5 continuous variables: age, resting blood pressure, serum cholestoral (in mg/dl), maximum heart rate achieved, and ST depression induced by exercise relative to rest (oldpeak); and 8 categorical variables: sex, chest pain type, fasting blood sugar $>$ 120 mg/dl (true or false), resting electrocardiographic results, exercise induced angina (yes or no), the slope of the peak exercise ST segment, number of major vessels (0-3) colored by flourosopy, and thal (normal, fixed defect, reversible defect). Figure 1

Figure 1: Summary of Variables

skimpy summary

Data Summary

dataframe	Values
Number of rows	1025
Number of columns	14

Data Types

Column Type	Count
int32	13
float64	1

number

column_name	NA	NA %	mean	sd	p0	p25	p75	p100	hist
age	0	0	54	9.1	29	48	61	77	▁▂▅▇▆▂▁
sex	0	0	0.7	0.46	0	0	1	1	▃▁▁▁▁▁▁▇
cp	0	0	0.94	1	0	0	2	3	▇▂▁▂▁▁▁▂
trestbps	0	0	130	18	94	120	140	200	▂▅▇▃▂▁▁▁
chol	0	0	250	52	130	210	280	560	▁▃▇▃▂▁▁▁
fbs	0	0	0.15	0.36	0	0	0	1	▇▁▁▁▁▁▁▂
restecg	0	0	0.53	0.53	0	0	1	2	▇▁▇▁▁▁▁▇
thalach	0	0	150	23	71	130	170	200	▁▂▃▅▇▆▃▂
exang	0	0	0.34	0.47	0	0	1	1	▇▁▁▁▁▁▁▃
oldpeak	0	0	1.1	1.2	0	0	1.8	6.2	▇▃▂▁▁▁▁▁
slope	0	0	1.4	0.62	0	1	2	2	▁▁▇▁▁▇▁▇
ca	0	0	0.75	1	0	0	1	4	▇▃▁▂▁▁▁▁
thal	0	0	2.3	0.62	0	2	3	3	▁▁▁▇▁▇▇▇
target	0	0	0.51	0.5	0	0	1	1	▇▁▁▁▁▁▁▇

End

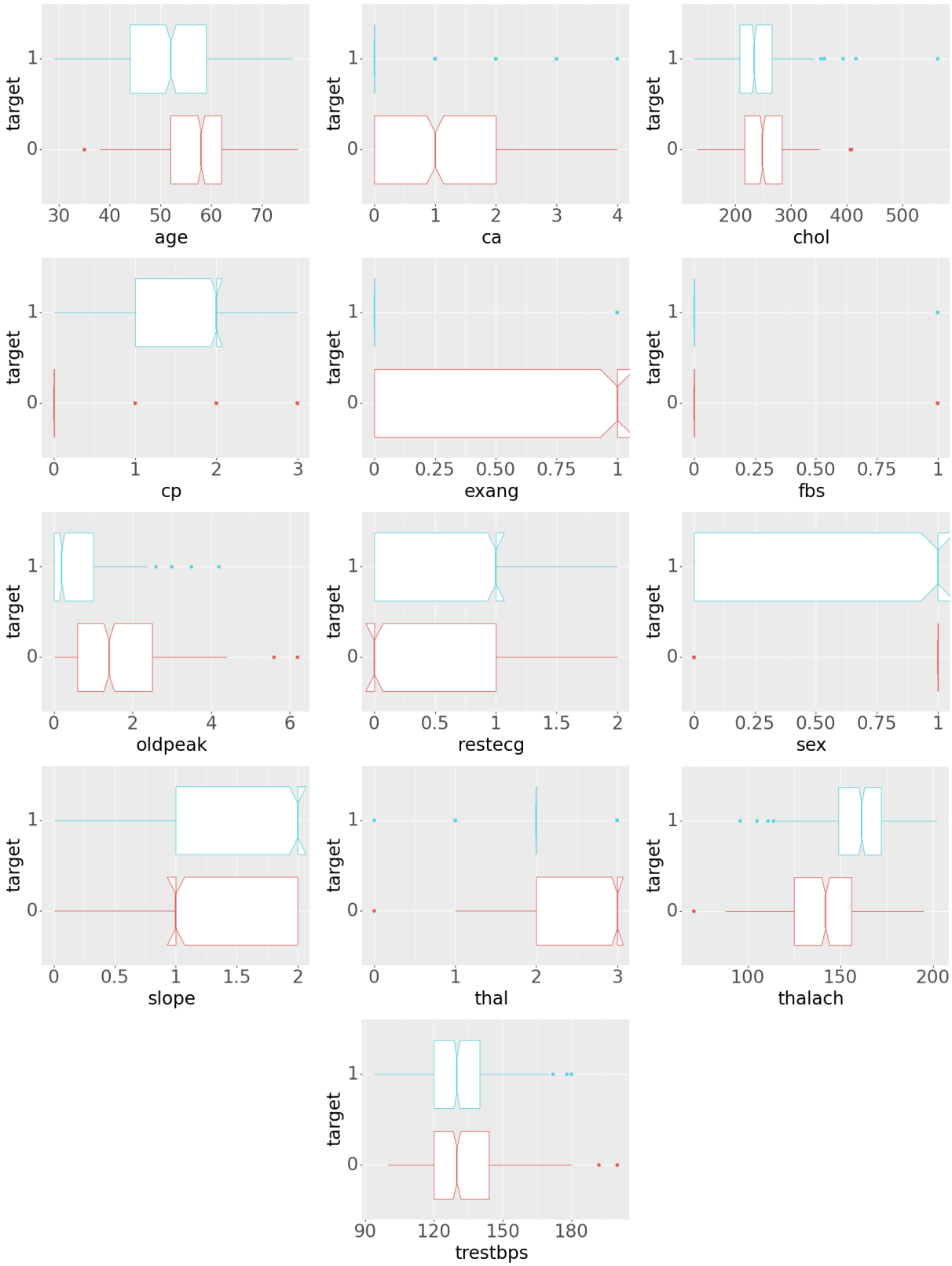


Figure 2: Distributions of Feature Variables with Respect to the Target

shows the `skimpy` summary of all 14 variables and figure 2 shows the distributions of predictor variables when compared to the target, a binary indicator for the patient having heart disease.

Prior to computing any of the above measures of associated risk, the data was imported into Python (3.9) using `pandas` (1.5.2) and dummy variables were created for the categorical columns. Individual risk differences and risk ratios were then computed for each predictor variable. The dataset was then divided into a 3 : 2 train-test split so that a logistic regression model can be fit. The models was fit using the training set to the objective function,

$$\text{logit}\,(Y) = \vec{\beta} X$$

via `statsmodels` (0.13.5). The odds ratios and the marginal effects were computed for each feature using the outputs of the logistic regression model. The model was then tested with the remainder of the data and scored using `scikit-learn` (1.1.3).

4 Results

Table 2 shows the risk difference and the relative risk of feature variables when compared against the target. The variable with the highest risk difference is *thalassemia = 2*, with a difference of 0.5203; meaning, on average, in a sample of 100 patients 52 more patients who have Thalassemia (a blood disease characterized by low hemoglobin production) as a fixed defect will develop heart disease than patients without Thalassemia. The variable with the lowest risk difference is is *thalassemia = 3* with a difference of -0.4894; meaning, on average, in a sample of 100 patients 49 fewer patients who have Thalassemia as a reversible defect will develop heart disease than patients without Thalassemia. Comparing the risk difference to the risk ratio, variables with risk

	Risk Difference	Risk Ratio	RR 95% Confidence Interval
age	-0.2892	0.5671	(0.5008, 0.6423)*
sex	-0.3036	0.5809	(0.5204, 0.6484)*
resting blood pressure $>$ 130	-0.0888	0.839	(0.7411, 0.9499)*
serum cholesterol $>$ 250 ml/dl	-0.1512	0.7374	(0.6473, 0.8401)*
fasting blood sugar $>$ 120 mg/dl	-0.0577	0.8893	(0.7415, 1.0666)
maximum heart rate achieved $>$ 150	0.4067	2.364	(2.0403, 2.7391)*
exercise induced angina	-0.4633	0.3076	(0.2483, 0.3809)*
oldpeak	-0.3773	0.435	(0.3708, 0.5103)*
chest pain = 1	0.3455	1.7563	(1.5815, 1.9503)*
chest pain = 2	0.3568	1.8613	(1.6732, 2.0704)*
chest pain = 3	0.1613	1.3219	(1.1134, 1.5694)*
resting electrocardiograph = 1	0.1785	1.4212	(1.2566, 1.6073)*
resting electrocardiograph = 2	-0.3178	0.3862	(0.1401, 1.0646)
slope = 1	-0.3499	0.4837	(0.4203, 0.5566)*
slope = 2	0.3904	2.167	(1.9032, 2.4674)*
major vessels colored = 1	-0.2837	0.5073	(0.4105, 0.6268)*
major vessels colored = 2	-0.4101	0.2765	(0.1859, 0.4112)*
major vessels colored = 3	-0.4104	0.2412	(0.1308, 0.4448)*
major vessels colored = 4	0.3259	1.6422	(1.324, 2.0369)*
thalassemia = 1	-0.1974	0.6244	(0.4375, 0.8911)*
thalassemia = 2	0.5203	3.1955	(2.7033, 3.7772)*
thalassemia = 3	-0.4894	0.3096	(0.2562, 0.3742)*

Table 2: Risk Difference and Risk Ratio of Binary Variables

differences less than zero have risk ratios less than 1 and variables with risk differences greater than zero have risk ratios greater than 1. Of the 22 risk ratios, only 2 variables have confidence intervals that contain 1, and thus accept the null-hypothesis. The variable with the highest statistically significant risk ratio is *thalassemia = 2*, with a risk ratio of 3.1955; meaning, patients who have Thalassemia as a fixed defect are 3.1955 times more likely to develop heart disease than patients without Thalassemia. The variable with the lowest statistically significant risk ratio is *major vessels colored = 3*, with a risk ratio of 0.2412; meaning, patients who had 3 major blood vessels colored in a flourosopy are 0.2412 times less likely to develop heart disease than patients who had no major blood vessels colored in a flourosopy.

Dep. Variable:	target	**No. Observations:**	615
Model:	Logit	**Df Residuals:**	593
Method:	MLE	**Df Model:**	21
Date:	Mon, 28 Nov 2022	**Pseudo R-squ.:**	0.5868
Time:	20:02:19	**Log-Likelihood:**	-175.99
converged:	True	**LL-Null:**	-425.93
Covariance Type:	nonrobust	**LLR p-value:**	1.574e-92

	coef	**std err**	**z**	**P> |z|**	**[0.025**	**0.975]**
age	0.0376	0.018	2.080	0.038	0.002	0.073
sex	-2.2499	0.433	-5.195	0.000	-3.099	-1.401
resting blood pressure	-0.0244	0.009	-2.857	0.004	-0.041	-0.008
serum cholesterol	-0.0145	0.004	-3.853	0.000	-0.022	-0.007
fasting blood sugar $>$ 120 mg/dl	0.2915	0.441	0.661	0.509	-0.573	1.156
maximum heart rate achieved	0.0242	0.008	2.953	0.003	0.008	0.040
exercise induced angina	-0.4887	0.321	-1.523	0.128	-1.118	0.140
oldpeak	-0.5662	0.177	-3.206	0.001	-0.912	-0.220
chest pain = 1	0.9770	0.421	2.320	0.020	0.152	1.802
chest pain = 2	1.8772	0.380	4.942	0.000	1.133	2.622
chest pain = 3	2.4143	0.497	4.854	0.000	1.439	3.389
resting electrocardiograph = 1	0.1241	0.286	0.434	0.664	-0.436	0.685
resting electrocardiograph = 2	-0.9711	3.222	-0.301	0.763	-7.286	5.344
slope = 1	-0.6824	0.610	-1.118	0.264	-1.879	0.514
slope = 2	0.8047	0.657	1.224	0.221	-0.484	2.093
major vessels colored = 1	-2.4127	0.394	-6.126	0.000	-3.185	-1.641
major vessels colored = 2	-3.3407	0.553	-6.041	0.000	-4.425	-2.257
major vessels colored = 3	-2.3139	0.697	-3.319	0.001	-3.680	-0.947
major vessels colored = 4	1.2003	1.213	0.990	0.322	-1.177	3.577
thalassemia = 1	4.1461	1.976	2.098	0.036	0.273	8.019
thalassemia = 2	3.9308	1.967	1.999	0.046	0.076	7.785
thalassemia = 3	2.6183	1.976	1.325	0.185	-1.255	6.492

Table 3: Logistic Regression Results

To compute the odds ratios and the marginal effects, a logistic regression model was fit. Table 3 shows the regression results using the maximum likelihood method. The log-likelihood of the model is -175.99, which is significantly larger than the log-likelihood of the null-model, -425.93, given the log-likelihood ratio test p-value of $1.574\text{e}{-92}$.

Table 4 shows the odds ratios of the regression, which were obtained by exponentiating the coefficients in table 3. Of the 22 odds ratios, 8 variables have confidence intervals that contain 1, and thus accept the null-hypothesis. The variable with the highest statistically significant odds ratio

	Odds Ratio	OR 95% Confidence Interval
age	1.0383	(1.0022, 1.0758)*
sex	0.1054	(0.0451, 0.2463)*
resting blood pressure	0.9759	(0.9597, 0.9924)*
serum cholesterol	0.9856	(0.9784, 0.9929)*
fasting blood sugar $>$ 120 mg/dl	1.3384	(0.5639, 3.1764)
maximum heart rate achieved	1.0245	(1.0082, 1.0411)*
exercise induced angina	0.6134	(0.3271, 1.1505)
oldpeak	0.5677	(0.4016, 0.8025)*
chest pain = 1	2.6565	(1.1638, 6.0638)*
chest pain = 2	6.5352	(3.104, 13.7594)*
chest pain = 3	11.1815	(4.2184, 29.6385)*
resting electrocardiograph = 1	1.1322	(0.6464, 1.9829)
resting electrocardiograph = 2	0.3787	(0.0007, 209.2937)
slope = 1	0.5054	(0.1528, 1.6717)
slope = 2	2.2360	(0.6163, 8.1119)
major vessels colored = 1	0.0896	(0.0414, 0.1938)*
major vessels colored = 2	0.0354	(0.012, 0.1047)*
major vessels colored = 3	0.0989	(0.0252, 0.3878)*
major vessels colored = 4	3.3210	(0.3082, 35.7816)
thalassemia = 1	63.1862	(1.3143, 3037.8507)*
thalassemia = 2	50.9487	(1.0792, 2405.3763)*
thalassemia = 3	13.7129	(0.285, 659.8043)

Table 4: Odds Ratios

is *thalassemia = 1*, with an odds ratio of 63.1862; meaning, the odds of contacting heart disease are 63.1862 times higher for patients with normal Thalassemia than patients without Thalassemia. The variable with the lowest statistically significant odds ratio is *major vessels colored = 2*, with an odds ratio of 0.0354; meaning, the odds of contacting heart disease are 0.0354 times lower for patients who had two major blood vessels colored by a fluoroscopy than patients who had no major blood vessels colored by a fluoroscopy.

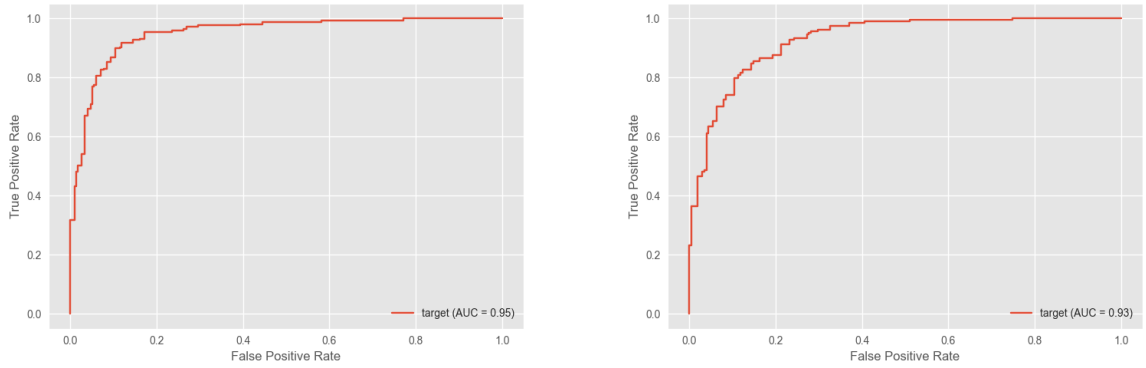
Table 5 shows the marginal effects of the regression output. Of the 22 marginal effects, 8 variables have p-values greater than 5%, and thus accept the null-hypothesis. It's important to note that these are the same variables that have statistically insignificant odds ratios. The variable with the highest statistically significant marginal effect is *thalassemia = 1*, with an average marginal effect of 0.3661; meaning that the average change in probability is 0.3661 when a patient has

Dep. Variable:	target					
Method:	dydx					
At:	overall					
	dy/dx	**std err**	**z**	**P> |z|**	**[0.025**	**0.975]**
age	0.0033	0.002	2.109	0.035	0.000	0.006
sex	-0.1986	0.035	-5.658	0.000	-0.267	-0.130
resting blood pressure	-0.0022	0.001	-2.915	0.004	-0.004	-0.001
serum cholesterol	-0.0013	0.000	-4.070	0.000	-0.002	-0.001
fasting blood sugar $>$ 120 mg/dl	0.0257	0.039	0.662	0.508	-0.050	0.102
maximum heart rate achieved	0.0021	0.001	3.038	0.002	0.001	0.004
exercise induced angina	-0.0431	0.028	-1.533	0.125	-0.098	0.012
oldpeak	-0.0500	0.015	-3.324	0.001	-0.079	-0.021
chest pain = 1	0.0863	0.036	2.378	0.017	0.015	0.157
chest pain = 2	0.1657	0.031	5.338	0.000	0.105	0.227
chest pain = 3	0.2132	0.041	5.210	0.000	0.133	0.293
resting electrocardiograph = 1	0.0110	0.025	0.434	0.664	-0.039	0.060
resting electrocardiograph = 2	-0.0857	0.284	-0.301	0.763	-0.643	0.472
slope = 1	-0.0602	0.054	-1.122	0.262	-0.166	0.045
slope = 2	0.0710	0.058	1.229	0.219	-0.042	0.184
major vessels colored = 1	-0.2130	0.030	-7.037	0.000	-0.272	-0.154
major vessels colored = 2	-0.2949	0.043	-6.804	0.000	-0.380	-0.210
major vessels colored = 3	-0.2043	0.059	-3.447	0.001	-0.320	-0.088
major vessels colored = 4	0.1060	0.107	0.991	0.322	-0.104	0.316
thalassemia = 1	0.3661	0.172	2.124	0.034	0.028	0.704
thalassemia = 2	0.3470	0.172	2.022	0.043	0.011	0.683
thalassemia = 3	0.2312	0.174	1.331	0.183	-0.109	0.572

Table 5: Marginal Effects

normal Thalassemia. The variable with the lowest statistically significant marginal effect is *major vessels colored = 2*, with an average marginal effect of -0.2949; meaning that the average change in probability is -0.2949 when a patient who has two major blood vessels colored by a fluoroscopy.

When predicting in sample, the model had a log-loss score of 0.2862, a Brier score of 0.0836, and an AUC of 0.9501, indicating that the model predicts well in sample. After training the model, a receiver operating characteristic analysis was applied to determine the optimal threshold for predicting the target. To calculate the ROC metrics, a confusion matrix was created for each



(a) In Sample (b) Out of Sample

Figure 3: Receiver Operator Curves

hundredth between 0 and 1. The selected threshold was 0.512 with an accuracy of 0.8943, a precision of 0.8941, a sensitivity of 0.9025, and a specificity of 0.8855. When predicting out of sample, the model had a log-loss score of 0.3406, a Brier score of 0.1073, and an AUC of 0.9294, indicating that the model does not lose much generality when introduced to new data. Figure 3 shows the in sample and out of sample receiver operator curves.

5 Discussion

Logistic regression is a useful tool in medical testing because it offers a means of binary classification while also providing insight into the measure of risk associated with each feature variable. Odds ratios and marginal effects can be calculated alongside a logistic regression model, which are often easier to interpret than methods like risk difference and risk ratio, as illustrated in section 4. Although the model is able to predict with a high success rate, the analysis did not consider other kinds of generalized linear models that can be used to estimate the associated risk, such as Poisson regression. Further research is needed to determine the best model to truly understand the

association between risk factors and developing heart disease.

References

Gallis, J. A. and Turner, E. L. (2019). Relative measures of association for binary outcomes: Challenges and recommendations for the global health researcher. *Ann Glob Health*, 85(1):137.

Lapp, D. (2019). Heart disease dataset.

Norton, E. C., Dowd, B. E., and Maciejewski, M. L. (2019). *Marginal Effects—Quantifying the Effect of Changes in Risk Factors in Logistic Regression Models*. McGraw-Hill Education, New York, NY.

Peng, C.-Y. J., Lee, K. L., and Ingersoll, G. M. (2002). An introduction to logistic regression analysis and reporting. *The Journal of Educational Research*, 96(1):3–14.

Schober, P. and Vetter, T. R. (2021). Logistic regression in medical research. *Anesthesia & Analgesia*, 132(2).

Telke, S. E. and Eberly, L. E. (2011). Statistical hypothesis testing: Associating patient characteristics with a prevalent or incident condition—relative risk, odds ratio, and logistic regression. *J. Wound Ostomy Continence Nurs.*, 38(5):496–500.

Wasserman, L. (2004). *Linear and Logistic Regression*, pages 209–229. Springer New York, New York, NY.

A Code

The code used in sections A.1 through A.3 can be found at https://github.com/TzipHoro/ACM725-LogisticRegression

A.1 data_summary.py

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from skimpy import skim


# read data
df = pd.read_csv('data/heart.csv')

X = df.drop('target', axis=1)
y = df['target']

corr = df.corr()


def get_lower_tri_heatmap(data, output="plots/correlation.png"):
    mask = np.zeros_like(data, dtype=np.bool)
    mask[np.triu_indices_from(mask)] = True

    # Want diagonal elements as well
    mask[np.diag_indices_from(mask)] = False

    # Set up the matplotlib figure
    f, ax = plt.subplots(figsize=(11, 9))

    # Generate a custom diverging colormap
    cmap = sns.diverging_palette(220, 10, as_cmap=True)

    # Draw the heatmap with the mask and correct aspect ratio
    sns_plot = sns.heatmap(data, mask=mask, cmap=cmap, vmax=.3, center=0,
            square=True, linewidths=.5, cbar_kws={"shrink": .5})
    # save to file
    fig = sns_plot.get_figure()
    fig.savefig(output)


def get_box_plots(data):
    import plotnine as gg

    for col in data:
        plot = gg.ggplot(df) +\
               gg.geom_boxplot(gg.aes(y=col, x='factor(target)', color='factor
   (target)'), notch=True) +\
               gg.xlab('target') +\
```

```
                gg.coord_flip() +\
                gg.scale_color_discrete(guide=False) +\
                gg.theme(text=gg.element_text(size=24))
        plot.save(f'plots/target-{col}.png', 'png')


if __name__ == '__main__':
    skim(df)
    # plot all sub-distributions
    get_box_plots(X)

    # corr plot
    get_lower_tri_heatmap(corr)
```

A.2 associated_risk.py

```
import warnings
import pandas as pd
from scipy.stats.contingency import relative_risk

from data_summary import X, y

warnings.simplefilter(action='ignore', category=FutureWarning)

# create dummies for categorical vars
X['thal'] = X['thal'].astype('category')
X['cp'] = X['cp'].astype('category')
X['restecg'] = X['restecg'].astype('category')
X['slope'] = X['slope'].astype('category')
X['ca'] = X['ca'].astype('category')
X = pd.get_dummies(X, drop_first=True)
X = pd.get_dummies(X, drop_first=True)

# individual cross-tabs
df = X.copy()
df['age'] = (X['age'] > 54).astype(int)
df['trestbps'] = (df['trestbps'] > 130).astype(int)
df['chol'] = (df['chol'] > 250).astype(int)
df['thalach'] = (df['thalach'] > 150).astype(int)
df['oldpeak'] = (df['oldpeak'] > 1.1).astype(int)

# calculate ARD and RR
ard = lambda de, du, he, hu: round((de / (de + he)) - (du / (du + hu)), 4)
risk_diff = dict()
risk_ratio = dict()
ci = dict()

for col in df.columns:
    crosstab = pd.crosstab(df[col], y)
    De = crosstab.iloc[1, 1]
    Du = crosstab.iloc[0, 1]
    He = crosstab.iloc[1, 0]
    Hu = crosstab.iloc[0, 0]

    risk_diff[col] = ard(De, Du, He, Hu)
    rr = relative_risk(De, De + He, Du, Du + Hu)
    risk_ratio[col] = round(rr.relative_risk, 4)
    ci[col] = rr.confidence_interval(confidence_level=0.95)


risk = pd.DataFrame.from_records([risk_diff, risk_ratio, ci],
                                 index=['Risk Difference', 'Risk Ratio', 'RR
    95% Confidence Interval'])
risk = risk.transpose()
risk['RR 95% Confidence Interval'] = risk['RR 95% Confidence Interval'].apply(
    lambda row: tuple([round(i, 4) for i in row]))
risk['RR 95% Confidence Interval'] = risk['RR 95% Confidence Interval'].apply(
    lambda row: str(row) + '*' if not (row[0] <= 1 <= row[1]) else str(row))
```

```
52
53 risk.set_axis(['age', 'sex', 'resting blood pressure > 130', 'serum
       cholesterol > 250 ml/dl',
54                 'fasting blood sugar > 120 mg/dl', 'maximum heart rate achieved
        > 150', 'exercise induced angina',
55                 'oldpeak', 'chest pain = 1', 'chest pain = 2', 'chest pain = 3'
       , 'resting electrocardiograph = 1',
56                 'resting electrocardiograph = 2', 'slope = 1', 'slope = 2', '
       major vessels colored = 1',
57                 'major vessels colored = 2', 'major vessels colored = 3', '
       major vessels colored = 4',
58                 'thalassemia = 1', 'thalassemia = 2', 'thalassemia = 3'],
       inplace=True)
59
60 print('RD:')
61 print(risk[risk['Risk Difference'] == risk['Risk Difference'].max()])
62 print(risk[risk['Risk Difference'] == risk['Risk Difference'].min()])
63 print('RR:')
64 print(risk[risk['Risk Ratio'] == risk['Risk Ratio'].max()])
65 print(risk[risk['Risk Ratio'] == risk['Risk Ratio'].min()])
66
67 risk.to_latex('src/risk-tables.tex', column_format='lrrr')
```

A.3 models.py

```
import sys
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import statsmodels.api as sm
from sklearn.model_selection import train_test_split
from sklearn.metrics import brier_score_loss, roc_auc_score, log_loss

from associated_risk import X, y

# import ROCMetrics
sys.path.insert(0, r'C:\Users\tzipo\Documents\GitHub\NeuralNetworks-FinalProject')
from ROC import ROCMetrics

plt.style.use('ggplot')


# train/test split
X.columns = ['age', 'sex', 'resting blood pressure', 'serum cholesterol', 'fasting blood sugar > 120 mg/dl',
             'maximum heart rate achieved', 'exercise induced angina', 'oldpeak', 'chest pain = 1', 'chest pain = 2',
             'chest pain = 3', 'resting electrocardiograph = 1', 'resting electrocardiograph = 2', 'slope = 1',
             'slope = 2', 'major vessels colored = 1', 'major vessels colored = 2', 'major vessels colored = 3',
             'major vessels colored = 4', 'thalassemia = 1', 'thalassemia = 2', 'thalassemia = 3']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=42)

# logistic regression
log_reg = sm.Logit(y_train, X_train).fit()
with open('src/logistic-regression.tex', 'w') as f:
    f.write(log_reg.summary().as_latex())

# calculate odds ratios:
odds_ratios = pd.DataFrame(
    {
        "Odds Ratio": log_reg.params,
        "Lower CI": log_reg.conf_int()[0],
        "Upper CI": log_reg.conf_int()[1],
    }
)
odds_ratios = np.exp(odds_ratios)
odds_ratios['OR 95% Confidence Interval'] = odds_ratios.apply(lambda row:
                                                              (round(row.iloc[1], 4), round(row.iloc[2], 4)),
                                                              axis=1)
odds_ratios.drop(["Lower CI", "Upper CI"], inplace=True, axis=1)
odds_ratios['Odds Ratio'] = odds_ratios['Odds Ratio'].round(4)
```

```
odds_ratios['OR 95% Confidence Interval'] = odds_ratios['OR 95% Confidence Interval'].apply(
    lambda row: str(row) + '*' if not (row[0] <= 1 <= row[1]) else str(row))
print(odds_ratios[odds_ratios['Odds Ratio'] == odds_ratios['Odds Ratio'].max()])
print(odds_ratios[odds_ratios['Odds Ratio'] == odds_ratios['Odds Ratio'].min()])
odds_ratios.to_latex('src/odds-ratios.tex')

# marginal effects
ame = log_reg.get_margeff(at='overall', method='dydx')
with open('src/marginal-effects.tex', 'w') as f:
    f.write(ame.summary().as_latex())

ame_df = ame.summary_frame()
print(ame_df[ame_df['dy/dx'] == ame_df['dy/dx'].max()])
print(ame_df[ame_df['dy/dx'] == ame_df['dy/dx'].min()])

# determine threshold
yhat_in_sample = log_reg.predict(X_train)
roc = ROCMetrics(y_train, yhat_in_sample)
thresholds = roc.threshold_matrix(step_size=0.001)
p_th = thresholds.loc[['sensitivity', 'specificity', 'precision', 'accuracy', 'f1_score'], :].sum().idxmax()
print(thresholds.loc[:, p_th].round(4))
roc.roc_plot(thresholds, 'plots/roc-is-1.png')

# scoring in sample
brier_score = brier_score_loss(y_train, yhat_in_sample)
auc_ = roc_auc_score(y_train, yhat_in_sample)
ll = log_loss(y_train, yhat_in_sample)

# predict
yhat_out_of_sample = log_reg.predict(X_test)
roc_oos = ROCMetrics(y_test, yhat_out_of_sample)
thresholds_oos = roc_oos.threshold_matrix(step_size=0.001)
p_th = thresholds_oos.loc[['sensitivity', 'specificity', 'precision', 'accuracy', 'f1_score'], :].sum().idxmax()
print(thresholds.loc[:, p_th].round(4))
roc_oos.roc_plot(thresholds, 'plots/roc-oos-1.png')

# scoring oos
brier_score_oos = brier_score_loss(y_test, yhat_out_of_sample)
auc_oos = roc_auc_score(y_test, yhat_out_of_sample)
ll_oos = log_loss(y_test, yhat_out_of_sample)
```

A.4 ROCMetrics

The ROC.py script that was imported in A.3 was imported from another project. The following is the code for the ROCMetrics class that was used. The full script can be found in my GitHub repository at https://github.com/TzipHoro/NeuralNetworks-FinalProject.

```
import warnings
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn.metrics import accuracy_score, roc_curve, auc, RocCurveDisplay,
    confusion_matrix, f1_score


warnings.simplefilter(action='ignore', category=FutureWarning)
warnings.simplefilter(action='ignore', category=pd.errors.PerformanceWarning)
plt.style.use('seaborn-v0_8')


class ROCMetrics:
    def __init__(self, y_true: pd.Series, p_pred: pd.Series):
        self.y_true = y_true
        self.p_pred = p_pred

    def conf_matrix(self, y_pred: pd.Series) -> np.array:
        return confusion_matrix(self.y_true, y_pred)

    @staticmethod
    def sensitivity(conf_matrix: np.array) -> float:
        tp = conf_matrix[1][1]
        fn = conf_matrix[1][0]

        with np.errstate(divide='ignore', invalid='ignore'):
            sens = np.true_divide(tp, tp + fn)
            if sens == np.inf:
                sens = 0

        return sens

    @staticmethod
    def specificity(conf_matrix: np.array) -> float:
        tn = conf_matrix[0][0]
        fp = conf_matrix[0][1]

        with np.errstate(divide='ignore', invalid='ignore'):
            spec = np.true_divide(tn, tn + fp)
            if spec == np.inf:
                spec = 0

        return spec

    @staticmethod
    def precision(conf_matrix: np.array) -> float:
        tp = conf_matrix[1][1]
```

```
        fp = conf_matrix[0][1]

        with np.errstate(divide='ignore', invalid='ignore'):
            prec = np.true_divide(tp, tp + fp)
            if prec == np.inf:
                prec = 0

        return prec

    @staticmethod
    def fall_out(conf_matrix: np.array) -> float:
        tn = conf_matrix[0][0]
        fp = conf_matrix[0][1]

        with np.errstate(divide='ignore', invalid='ignore'):
            fo = np.true_divide(fp, tn + fp)
            if fo == np.inf:
                fo = 0

        return fo

    def accuracy(self, y_pred: pd.Series) -> float:
        return accuracy_score(self.y_true, y_pred)

    def f1_score(self, y_pred: pd.Series) -> float:
        return f1_score(self.y_true, y_pred)

    def threshold_matrix(self, step_size: float) -> pd.DataFrame:
        if not (0 <= step_size <= 1):
            raise ValueError('step_size must be a valid probability')
        cols = np.arange(0, 1, step_size)
        matrix = pd.DataFrame()

        for i in cols:
            y_pred = self.p_pred >= i
            conf = self.conf_matrix(y_pred)
            sens = self.sensitivity(conf)
            spec = self.specificity(conf)
            prec = self.precision(conf)
            fo = self.fall_out(conf)
            accr = self.accuracy(y_pred)
            f1 = self.f1_score(y_pred)

            y_pred = y_pred.append(pd.Series({'sensitivity': sens, '
specificity': spec, 'precision': prec,
                                              'fall_out': fo, 'accuracy': accr
, 'f1_score': f1}))
            matrix[i] = y_pred

        return matrix

    def roc_plot(self, threshold_matrix: pd.DataFrame,  path: str = None):
        plt.clf()

```

```
        fpr = threshold_matrix.loc['fall_out', :]
        tpr = threshold_matrix.loc['sensitivity', :]
        roc_auc = auc(fpr, tpr)

        display = RocCurveDisplay(fpr=fpr, tpr=tpr, roc_auc=roc_auc,
    estimator_name=self.y_true.name)
        display.plot()

        plt.title('ROC Curve')

        if path is not None:
            plt.savefig(path)

    @staticmethod
    def metrics_plot(threshold_matrix: pd.DataFrame, path: str = None, xlim:
    tuple = None):
        plt.clf()

        threshold_matrix.loc['sensitivity', :].plot(legend=True)
        threshold_matrix.loc['specificity', :].plot(legend=True)
        threshold_matrix.loc['precision', :].plot(legend=True)
        threshold_matrix.loc['accuracy', :].plot(legend=True)
        threshold_matrix.loc['f1_score', :].plot(legend=True)

        plt.title('ROC Metrics')
        plt.xlabel('Threshold')
        plt.ylabel('Score')
        if xlim is not None:
            plt.xlim(xlim)
        plt.plot()

        if path is not None:
            plt.savefig(path)
```